1. Oracle provided predicate interface. This interface present in java.util.function package.

2. predicate Interface is functional interface which has only one abstract method.i.e test() and many default methods.

Interface Predicate<T>{

Public Boolean test(T t);

----

}

Predicate is functional interface and hence it can refers lambda expression.

Example:1

**import** java.util.function.Predicate;

**class** suku{

**public** **static** **void** main(String args[]) {

Predicate <Integer>p=(a)->{

**if**(a>50)

**return** **true**;

**else**

**return** **false**;

};

System.***out***.println(p.test(40));

System.***out***.println(p.test(60));

}

}

Output:

false

true

**Example-2:**

**class** Employee{

**int** eno;

String ename;

Employee(**int** eno,String ename){

**this**.eno=eno;

**this**.ename=ename;

}

}

**class** suku{

**public** **static** **void** main(String args[]) {

Predicate <Employee>p=(a)->{

**if**(a.eno>0)

**return** **true**;

**else**

**return** **false**;

};

Employee e1=**new** Employee(1,"suku");

Employee e2=**new** Employee(0,"sv");

System.***out***.println(p.test(e1));

System.***out***.println(p.test(e2));

}

}

Output:

-------

true

false

**Predicate Joins:**

It is possible to join predicates into single predicate by using following methods.

1. And().
2. Or()
3. Negate().

Example:-

**class** suku{

**public** **static** **void** main(String args[]) {

Predicate <Integer>p1=(a)->{

**if**(a>10)

**return** **true**;

**else**

**return** **false**;

};

Predicate <Integer>p2=(b)->{

**if**(b<20)

**return** **true**;

**else**

**return** **false**;

};

Predicate <Integer>p3=p1.and(p2);

Predicate <Integer>p4=p1.or(p2);

Predicate <Integer>p5=p1.negate();

System.***out***.println(p3.test(11));

System.***out***.println(p3.test(9));

System.***out***.println(p4.test(9));

System.***out***.println(p4.test(21));

System.***out***.println(p5.test(21));

System.***out***.println(p5.test(5));

}}

true

false

true

true

false

true